

Anhang A

Sequenzen-Programmierung

Im Rahmen dieser Arbeit wurden verschiedene Methoden (Sequenzen) auf dem 3-Tesla-Tomographen implementiert, u.a. zur Messung von k -Raum-Trajektorien (s. Anhang B und Abschnitt 3.2), die segmentierte, flukompensierte TurboGRASS Sequenz fr die Herzbildgebung (s. Kapitel 5) und die EPSI Sequenz fr die Thermosonden-Methode (s. Abschnitt 7.3). Mit der Aufrstung des Tomographen auf vier Empfangskanle wurde die Steuersoftware ParaVision Version 2.0.1 (inklusive XWIN-NMR Version 2.5) installiert. Diese Softwareversion ist fr den Mehrkanal-Empfang ntig, so da ein Teil der Herzbildgebung mit dieser Version durchgefhrt wurde. Da einerseits nicht alle Protokolle fr diese Version vorliegen und andererseits noch ltere Studien laufen, ist die vorhergehende Softwareversion (ParaVision 1.1-5 mit XWIN-NMR 1.0) weiterhin vorhanden, mit der der grte Teil der Messungen in dieser Arbeit durchgefhrt wurden. Die Programmierung von neuen Methoden fr beide Versionen unterscheidet sich kaum. Es bestehen vorwiegend fr den Mehrkanal-Empfang einige Erweiterungen. Die neue Methoden-Programmierung mit dem *Method Manager* wurde hingegen nicht verwendet.

A.1 Puls- und Gradientenprogramm

Der Ablauf einer Sequenz wird durch das Pulsprogramm gesteuert [Bru99]. Werden in einer Sequenz auch Gradienten geschaltet, wird zustzlich ein Gradientenprogramm bentigt, da durch spezielle Befehle mit dem Pulsprogramm synchronisiert wird. Die implementierte segmentierte, flukompensierte TurboGRASS Sequenz (s. Abbildung 5.11) wird im Folgenden als Beispiel kurz dargestellt. Ausgangspunkt waren zwei Sequenzen des Herstellers: `gefi.ppg` und `gefc_tomo.ppg`.

Pulsprogramm

```
#include <DBX.include>
    preset off
#include <MEDSPEC.include>
start,      5u          fq8b:f1
if(ACQ_trigger_enable) {
    10u          setnmr3|8
```

```

        10u          trigp1
        10u          setnmr3^8
    }
    if(IMND_USER_tagging) {
        10u          fq2:f1
        d8 gatepulse 1
        p1:sp1       ph0
        d4:ngrad
        d4:ngrad
        d8 gatepulse 1
        p2:sp1       ph0
        d4:ngrad
        d4:ngrad
        d8 gatepulse 1
        p1:sp1       ph0
        d4:ngrad
        1m          d4:ngrad
    }
    if(ACQ_trigger_enable) {
        d7
    }
    movie,          10u
    echo,           d4:ngrad
                  d8 gatepulse 1
    if(IMND_USER_rf_spoiling == No) {
        p0:sp0       ph0
        d4:ngrad
        d5          ph0
    }
    if(IMND_USER_rf_spoiling == Yes) {
        p0:sp0       ph4
        d4:ngrad
        d5          ph5
    }
    d4:ngrad
    d10:ngrad
    d4:ngrad
    d4:ngrad
    d4:ngrad:e     syrec
    d9:e           fqatofqb:f1
    if(IMND_USER_rf_spoiling == No) {
        aqq         adc ph0
    }
    if(IMND_USER_rf_spoiling == Yes) {
        aqq         adc
    }
    10u            eoscp
    2.5u           ipp4
    2.5u           ipp5
    if(IMND_USER_grad_refo == No) {
        10u         tofqa:f1
    }
    if(IMND_USER_grad_refo == Yes) {
        d4:ngrad
        d5          tofqa:f1
    }
    lo to echo times NECHOES
    d12:ngrad
    lo to movie times L[20]
    d0            rpp0
    lo to start times NSLICES
    2.5u
    lo to start times NA
    2.5u
    goto start ph0 = 0
ph4 = (360) 0 117 351 342 90 315 297 36 252 225 315 162 126 207 45 0

```

```

72 261 207 90 27 81 252 180 225 27 306 342 135 45 72 216
117 135 270 162 171 297 180 180 297 171 162 270 135 342 306 27
225 180 252 81 27 90 270 207 261 72 0 45 207 126 162 315
225 252 36 297 315 90 342 351 117 0 0 117 351 342 90 315
297 36 252 225 315 162 126 207 45 0 72 261 207 270 90 27
81 252 180 225 27 306 342 135 45 72 216 117 135 270 162 171
297 180 180 297 171 162 270 135 117 216 72 45 135 342 306 27
225 180 252 81 27 90 270 207 261 72 0 45 207 126 162 315
225 252 36 297 315 90 342 351 117 0 0 351 342 90 315 297
36 252 225 315 162 126 207 45 0 72 261 207 270 90 27 81
252 180 225 27 306 342 135 45 72 216 117 135 270 162 171 297
180 180 297 171 162 270 135 117 216 72 45 135 342 306 27 225
180 252 81 27 90 270 207 261 72 0 45 207 126 162 315 225
252 36 297 315 90 342 351 117 0 0 117 351 342 90 315 342
ph5 = (360) 0 117 351 342 90 315 297 36 252 225 315 162 126 207 45 0
72 261 207 90 27 81 252 180 225 27 306 342 135 45 72 216
117 135 270 162 171 297 180 180 297 171 162 270 135 342 306
225 180 252 81 27 90 270 207 261 72 0 45 207 126 162 315
225 252 36 297 315 90 342 351 117 0 0 117 351 342 90 315
297 36 252 225 315 162 126 207 45 0 72 261 207 270 90 27
81 252 180 225 27 306 342 135 45 72 216 117 135 270 162 171
297 180 180 297 171 162 270 135 117 216 72 45 135 342 306 27
225 180 252 81 27 90 270 207 261 72 0 45 207 126 162 315
225 252 36 297 315 90 342 351 117 0 0 351 342 90 315 297
36 252 225 315 162 126 207 45 0 72 261 207 270 90 27 81
252 180 225 27 306 342 135 45 72 216 117 135 270 162 171 297
180 180 297 171 162 270 135 117 216 72 45 135 342 306 27 225
180 252 81 27 90 270 207 261 72 0 45 207 126 162 315 225
252 36 297 315 90 342 351 117 0 0 117 351 342 90 315 342

```

Gradientenprogramm

```

loop ACQ_size[1]/ACQ_rare_factor <2d> <rare>
{
  begin dummy_projection
  loop NA
  {
    loop NSLICES <slice>
    {
      if(IMND_USER_tagging)
      {
        { (t7) | (0) | (0) }
        { (0) | (0) | (0) }
        { (t7) | (0) | (0) }
        { (0) | (0) | (0) }
        { (25,,no_scale) | (25,,no_scale) | (25,,no_scale) }
        { (0) | (0) | (0) }
      }
      loop L[20]
      {
        loop NECHOES <rare>
        {
          { (0) | (0) | (100) }
          { (t4) | (0),r2d(t2) | (0) }
          { (0) | (0) | (t1) }
          { (t0) | (0),r2d(t6) | (t1) }
          { (t0) | (0),r2d(t6) | (0) }
          { (0) | (0) | (t5) }
          { (90) | (0) | (0) }
          if(IMND_USER_grad_refo == Yes)
          {
            { (0) | (0),r2d(t3) | (0) }
          }
        }
      }
    }
  }
}

```

```

    }
  {      (0)      | (0)      | (0)      }
    }
  }
}
end dummy_projection
}

```

Der Ablauf der Sequenz wird durch die ACQ (*ACQ*uision) Parameter (s. nächster Abschnitt, z.B. ACQ_dim, ACQ_size, ACQ_rare_factor, NECHOES, NSLICES und NA) bestimmt, die die Schleifenstruktur und somit die Reihenfolge der Akquisition steuern. Die Reihenfolge der Phasenkodierschritte wird durch die beiden Parameter ACQ_phase_encoding_mode und ACQ_phase_enc_start festgelegt und durch die lineare Rampenfunktion r2d im Gradientenprogramm realisiert. Dabei wird die segmentierte Ausführung durch die Deklaration der entsprechenden Schleifen mit dem Zusatz <rare> erreicht. Die Synchronisation des Puls- mit dem Gradientenprogramm erfolgt über den Zusatz :ngrad in dem Pulsprogramm.

Die Sequenz erlaubt:

1. Die Triggerung auf die R-Zacke des EKG. Diese wird von der TCU gesteuert. Der Befehl setnmr3|8 schaltet den Modus ein, so daß mit dem Befehl trigpl1 auf einen positiven Wert in dem entsprechenden Register gewartet wird, um dann den Modus wieder durch setnmr3^8 abzuschalten. Das Triggersignal kommt dabei von dem EKG Überwachungsmonitor (Physiogard SM 785 NMR), kann aber auch von einer anderen Quelle stammen (s. Abschnitt 4.2.3).
2. 1–2–1 SPAMM Tagging.
3. *RF phase spoiling* mit einem Phaseninkrement von 117° [Zur91] (ph4 und ph5).
4. Gradientenrefokussierung in Phasenkodierrichtung (GRASS, Standardeinstellung).
5. Cine-Bildgebung (lo to movie times L[20]).

Da die Amplitude der Anregungspulse Bestandteil der Pulsparameter (z.B. *shaped pulse* sp0) ist, sind variable Drehwinkel, d.h. eine Veränderung des Drehwinkels eines Pulses während des Experimentes entsprechend einer vorher berechneten Liste, nicht möglich (s. Abschnitt 4.1.3).

Die Parameter d0 bis d12 entsprechen den Parametern D[0] bis D[12] des ACQP (d0= 410 μ s, d5= 180 μ s, d8= 100 μ s, d10 \geq 410 μ s). Die Auslesezeit aqq betrug 256/90 kHz \approx 2,8 ms und die Pulslänge p0= 500 μ s.

A.2 ACQP und IMND

Eine Vielzahl von Parametern sind zur Steuerung einer Sequenz nötig und sind im ACQP zusammengefaßt. Auch wenn diese weder Bestandteil des Puls- noch des Gradientenprogrammes

sind, regeln diese z.B. die ausgewählte Schicht (*ACQ_grad_matrix*), den Digitalisierungsmodus (*AQ_mod*) oder den Ausgleich der Laufzeit-Verschiebung (*group delay*) der digitalen Filter (*ACQ_scan_shift*, ab ParaVision 2.0, s. Abschnitt 3.3.2). Diese Parameterebene ist ausreichend zur Steuerung des Tomographen.

Für eine anwenderfreundlichere Bedienung, die auch die Änderung sequenzspezifischer Parameter erlaubt, wurde eine weitere Parameterebene eingeführt, und zwar die IMND (*IMaging N-Dimensional*) Parameter. Auf dieser Ebene können eigene Methoden mit neuen Parametern erstellt und in das vorhandene IMND-*Overlay* eingegliedert werden. Aus diesen Parametern werden dann, teilweise durch vorhandene Funktionen, die Parameter des ACQP berechnet. Um das GUI (*Graphics' User Interface*) mit dem *Geometry Editor* verwenden zu können, ist eine Methoden-Programmierung auf der IMND Parameterebene ratsam.

Beim Starten einer Sequenz werden zuerst Puls- und Gradientenprogramm mit den Parametern des ACQP kompiliert, sowie einige weitere, teilweise ausschaltbare Prozesse gestartet. Dies dauert abhängig von der Komplexität der Programme unterschiedlich lange, so daß zwischen dem Zeitpunkt des Betätigens des *GOP Buttons* und dem eigentlichen Starten der Sequenz einige Sekunden (bis über eine Minute bei einer Sequenz, die eigene Gradienten- und HF-Pulsformen verwendet) vergehen können.